RL-TR-94-228
Final Technical Report
December 1994

AUTOMATED OPTICAL TARGET RECOGNITION

Florida Institute of Technology

Samuel P. Kozaitis

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-94-228 has been reviewed and is approved for publication.

APPROVED: Wesley E. Foor

WESLEY E. FOOR
Project Engineer

FOR THE COMMANDER: Donald W. Hanson

DONALD W. HANSON
Director of Surveillance & Photonics

RL-TR-94-228
Final Technical Report
December 1994

AUTOMATED OPTICAL TARGET RECOGNITION

Florida Institute of Technology

Samuel P. Kozaitis

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
December 1994

3. REPORT TYPE AND DATES COVERED
Final Dec 93 - Sep 94

4. TITLE AND SUBTITLE

AUTOMATED OPTICAL TARGET RECOGNITION

5. FUNDING NUMBERS
C - F30602-94-C-0008
PE - 62702F
PR - 4600
TA - P1
WU - PX

6. AUTHOR(S)

Samuel P. Kozaitis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Florida Institute of Technology
Division of Electrical & Computer Science & Engineering
150 W. University Blvd
Melbourne FL 32901-6988

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (OCPA)
25 Electronic Pky
Griffiss AFB NY 13441-4515

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-94-228

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Wesley E. Foor/OCPA/(315) 330-2944

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report presents an overview of continuous and discrete wavelet transforms. Both digital and optical implementations of the discrete wavelet transform are discussed. Examples of typical wavelet basis functions are compared and the constraints imposed by optical implementations are discussed. A multi-resolution signal processing approach to object recognition is presented using an optical correlator for generating a wavelet transform. The practicality of using an optical correlator to perform target recognition with a wavelet representation approach is considered.

14. SUBJECT TERMS
Wavelet transform, Optical correlation, Multi-resolution processing, Target recognition

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1.0 Introduction

There has been much research into wavelets as applied to signal and image processing in recent years. Many papers and books have been published on wavelets and their applications including an interesting historical perspective.[1] Many of the papers involve time-frequency or scale-frequency analysis, and multiresolution processing but there are many others.[2-4] We've attempted to collect the most important aspects of wavelets and present them in a manner that is easy to understand with an emphasis on optical correlation processing.

Like the Fourier transform (FT), the wavelet transform (WT) describes a function with basis functions. However, the basis functions of the WT, wavelets, are often more complicated than the basis functions of the FT, sines and cosines. Furthermore, unlike the FT the basis functions in the WT are localized in both the input and wavelet domain. This dual localization has some important consequences. One is that many functions and operators can be represented quite sparsely in the wavelet domain. In addition, noise may be reduced because it is generally not localized in the wavelet domain and it may be easy to separate the noise from a signal.

Like the FT, the WT is a linear operation which is invertible and can be made orthogonal. In its orthogonal form which is used for most applications, the WT operates on vectors whose length are a power of two. The size of the vectors in the wavelet domain remains the same as in the input domain. There is not a unique set of wavelets so care must be taken when drawing conclusions obtained with one set of wavelets.

We presented an overview of the WT and its implementation including digital and optical approaches. We considered the main points of the WT without relying heavily on mathematics in attempt to achieve a better understanding. Next, we considered multiresolution approximations using the WT. Finally, we presented a multiresolution processing example using an optical correlator for object recognition.

2.0 Wavelet transform

We described the foundations of the WT, first by describing the conditions required for a wavelet basis, then the continuous and discrete WT. We then described more practical considerations including the implementation of the discrete WT using moving-average filters and a algorithm suitable for programming. Finally, we described the two-dimensional WT.

2.1 General considerations

For a function to be a wavelet it must be admissible. In short, admissible functions are those that oscillate and have a fast decay, have finite energy, and zero mean. These signals are essentially bandpass signals and include a large class of functions. A function $h(t)$ is admissible if

$$C_h = \int_{-\infty}^{\infty} \frac{|H(\omega)|^2}{|\omega|} d\omega < \infty \qquad (1)$$

where $H(\omega)$ is the FT of $h(t)$. For multiresolution, orthogonal, or biorthogonal wavelet transforms, wavelets must satisfy additional constraints.

The decision on which wavelet to use can be made by considering their performance. Different wavelets may perform differently on a particular function. A wavelet that works well for compression may not work as well for analysis.

2.2 Continuous WT

The WT maps a signal in the time domain into a scale-translation domain using scaled and translated versions of a wavelet. The original wavelet, where scaled and translated versions are made from is called the mother wavelet. The WT of function $f(x)$ with respect to an admissible mother wavelet $h(x)$ is

$$W_h f(a, b) = \frac{1}{\sqrt{|a|}}\int f(x)\, h^* \left(\frac{x-b}{a}\right) dx. \qquad (2)$$

The wavelet $h_{ab}(x) = h(x - b)/a$ is a version of the mother wavelet that has been scaled by a factor of a, and translated by a factor of b. The factor multiplied by the integral is used to normalize energy so that all scaled wavelets have the same energy as the original mother wavelet. The freedom of choosing a mother wavelet makes general WT characterizations of a particular function difficult.

In the WT the input function $f(x)$ is being compared to a wavelet $h_{ab}(x)$ through a correlation or projection. A wavelet domain coefficient is computed for each particular scale and translation value a and b. The value of the wavelet coefficient corresponds to the degree of correlation between $f(x)$ and $h_{ab}(x)$. Determining these coefficients is the WT. In other words, a function is approximated by a weighted sum of the scaled and translated mother wavelet. Each wavelet acts as a building block and when summed together the original signal is obtained.

In addition, the WT is linear and superposition holds. If a function can be described by a sum of separate signals, then the WT of the function is the sum of the WT of each of the separate signals.

2.2.1 Inverse continuous WT

The inverse WT creates the original function by summing weighted, scaled and translated versions of the mother wavelet. The weights are the wavelet coefficients $W_h f(a,b)$. The inverse WT sums over the two -dimensional scale-translation space as

$$f(x) = \frac{1}{C_h \sqrt{|a|}}\iint W_h f(a, b)\, h\left(\frac{x-b}{a}\right) \frac{db\,da}{a^2}, \qquad (3)$$

where the constant C_h is often dropped. The reconstructing elements are the scaled and translated wavelets, but they have not been complex conjugated. If the WT was not taken with respect to the same mother wavelet, then the inverse transform with respect to a different mother wavelet will not necessarily reconstruct a function that resembles the original.

2.3 Discrete wavelet transform

Analogous to the FT, the discrete WT is usually implemented rather than the continuous WT. The continuous WT is redundant and Meyer[1] showed that there exists wavelets which are not redundant and form an orthonormal basis. The discrete WT is analogous to the Fourier series rather than the discrete FT. The scales and translations are discrete, not necessarily the function to be transformed. The wavelet coefficients can be thought of as corresponding to discrete points on a a two-dimensional grid in the scale-translation domain. Usually this grid is indexed by two integers m and n. In the discrete WT, the scale becomes $a = a_0{}^m$ where a_0 is the discrete scale step size. The translation b becomes $b = nb_0a_0{}^m$ where $b_0a_0{}^m$ is the discrete translation step size.

The discrete WT is defined with respect to a continuous mother wavelet and can be written as

$$W_hf(m,n) = a_0^{\frac{-m}{2}} \int f(x)\, h\,(a_0^{-m}x - nb_0)\, dx. \tag{4}$$

Note that the processing is over continuous time but the wavelet representation is on a discrete grid.

The time-scale or shift-scale characteristics of a function can be described in terms of a scale-translation lattice. The scale and translation discrete step sizes a_0 and $b_0\, a_0{}^m$determine the density of the discrete lattice. The translation step size $b_0\, a_0{}^m$controls how far a wavelet may be translated before the next wavelet coefficient is computed. The choice of $b_0\, a_0{}^m$will depend upon the time resolution properties of the mother wavelet. Often the scale discrete step size a_0 is set to 2, so that $a = 2^m$, and sampled signals are easily scaled by downsampling.

A particular wavelet can be thought of as a bandpass filter. At high frequencies (small scales) the bandwidth of the wavelet is large. When considering lower frequencies (larger scales) the bandwidth of the wavelet decreases. The values of $a = 2^m$, and $b = nb_02^m$ correspond to a time-frequency or time-scale lattice as shown in Fig. 1. Each cell in Fig. 1 corresponds to a correlation between a wavelet and the original function. Each cell represents the simultaneous time-scale resolution of each discrete wavelet. In Fig. 1 the scale is increasing in the downward direction while the frequency increases in the upward direction. At higher frequencies the bandwidth (width of a resolution cell) increases maintaining a constant Q (ratio) between center frequency and bandwidth. For small scales the mother wavelet is compressed in time leading to small time support and good time resolution, but poor frequency resolution. For large scales the mother wavelet is compressed in frequency leading to small frequency support and good frequency resolution, but poor time resolution.

At a particular scale the resolution cells are all the same size. The cells increase in length along the frequency axis by powers of two as the frequency increases. At the same time, the cells decrease in width (time or translation axis) by a factor of two as the frequency increases.

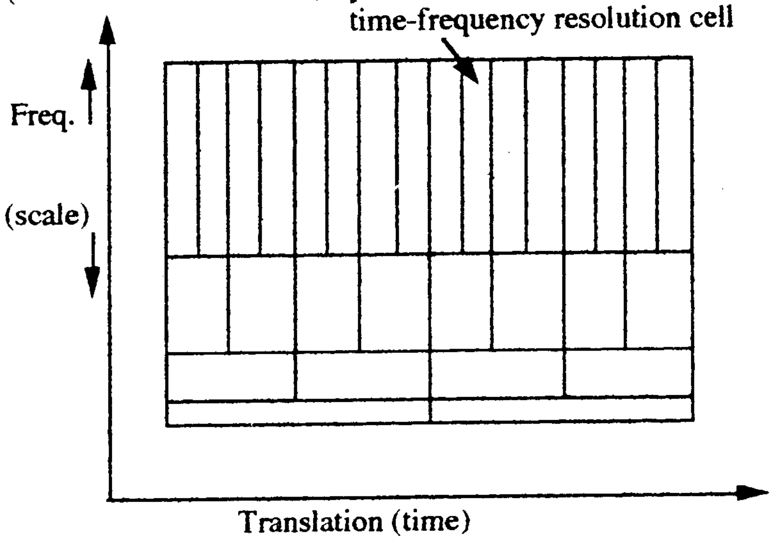


FIGURE 1. Scale-translation resolution of discrete WT

Each resolution cell corresponds to a discrete wavelet coefficient where each coefficient can be considered as the output of a correlation between a wavelet at a particular scale and translation, and the input signal.

Usually sampled signals are considered when implementing the discrete WT on a computer. We considered a general example in the frequency domain of how an *N*-byte signal was wavelet transformed. This decomposition is called the discrete time wavelet series because both the input and wavelet domains are discrete; however, we refer to it as the discrete WT as do most texts. According to the scale-translation grid in Fig. 1 the number of wavelet coefficients needed at a particular scale decreases as the scale increases. Furthermore, as the scale decreases, the frequency bandwidth increases. For a signal represented by an *N*-byte vector, at the smallest scale (widest frequency band) we need *N*/2 coefficients. As the scale increases the frequency bandwidth decreases by a factor of two for each wavelet so we need half as many coefficients, *N*/4. This is generally repeated until we have only 2 bytes left in the last remaining frequency band. This situation is illustrated in Fig. 2 which may be considered as a side view of Fig. 1.

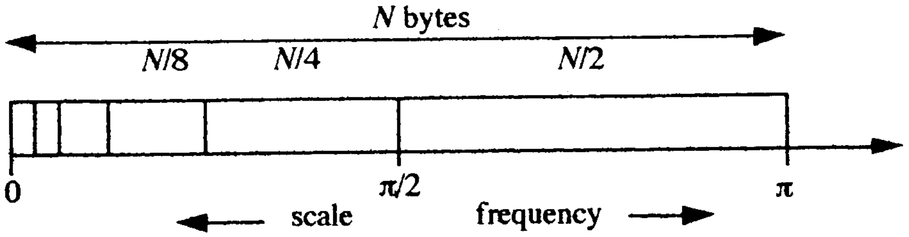


FIGURE 2. Frequency domain of *N*-byte signal

By examining Fig. 2 it can be seen that the wavelet acts as a bandpass filter. Therefore, in a general sense a wavelet can be thought of as a bandpass filter and generally does have complicated behavior.

Considering the grid in Fig. 1 the inverse discrete WT can be written as

$$f(x) = \sum\sum [W_g f(a,b)] a_0^{\frac{-m}{2}} h(a_0^{-m}x - nb_0) \tag{5}$$

where the summation is over m and n. The structure is the same as the continuous WT so the reconstruction processes are similar.

2.4 Implementation of an orthogonal WT with quadrature mirror filters

Usually, orthogonal or biorthogonal wavelets are used in wavelet representations with a scale step of 2. Orthogonal coefficients represent independent information so there is no redundancy. Orthogonal wavelets satisfy the condition that the cross-correlation between a scaled and translated mother wavelet and the mother wavelet is an impulse in scale and translation. This condition was written as

$$\int h_{m,n}(x) h_{m',n'}{}^{*}(x)\, dx = \delta(m-m')\delta(n-n'), \tag{6}$$

which is equal to 1 if m=m', and n=n', and equal to 0 otherwise. Independent wavelet coefficients are found by projecting an input function onto each basis element.

The wavelet transform is a hierarchial operator, it operates on one scale then a larger scale (lower resolution). Orthogonality requires that the information at each scale be independent of other scales. Because we will have two "branches" at each scale in our decomposition, if the two branches are orthogonal, our wavelet will be referred to as biorthogonal. Biorthogonal transforms relax some of the constraints on the mother wavelet and allows the mother wavelet to be symmetric and have linear phase.

The discrete WT can be implemented in an efficient manner by passing a signal through identically structured processing stages where each successive stage processes half the number of bits as the previous stage. In this configuration the first $N/2$ coefficients are generated at the first stage, $N/4$ at the next and so on. A schematic diagram of the discrete WT decomposition is shown in Fig. 3. The wavelet coefficients generated correspond to the resolution cells in Fig. 1. In general N - 1 processing blocks are needed to decompose an N - bit input.

The approximation of the signal $f(x)$ at the resolution 2^j is referred to as $A_{2^j}\{f(x)\}$ where A_{2^j} is a projection operator that approximates the function $f(x)$ and $j < 0$. The signal $A_1\{f(x)\}$ is $f(x)$, the original signal at the highest resolution, and $A_{1/2}\{f(x)\}$, $A_{1/4}\{f(x)\}$, etc. are lower resolution versions of $f(x)$. The detail signal $D_{2^j}\{f(x)\}$ at the resolution 2^j contains the difference of information between $A_{2^{j+1}}\{f(x)\}$ and $A_{2^j}\{f(x)\}$. These signals are indicated in Fig. 3.

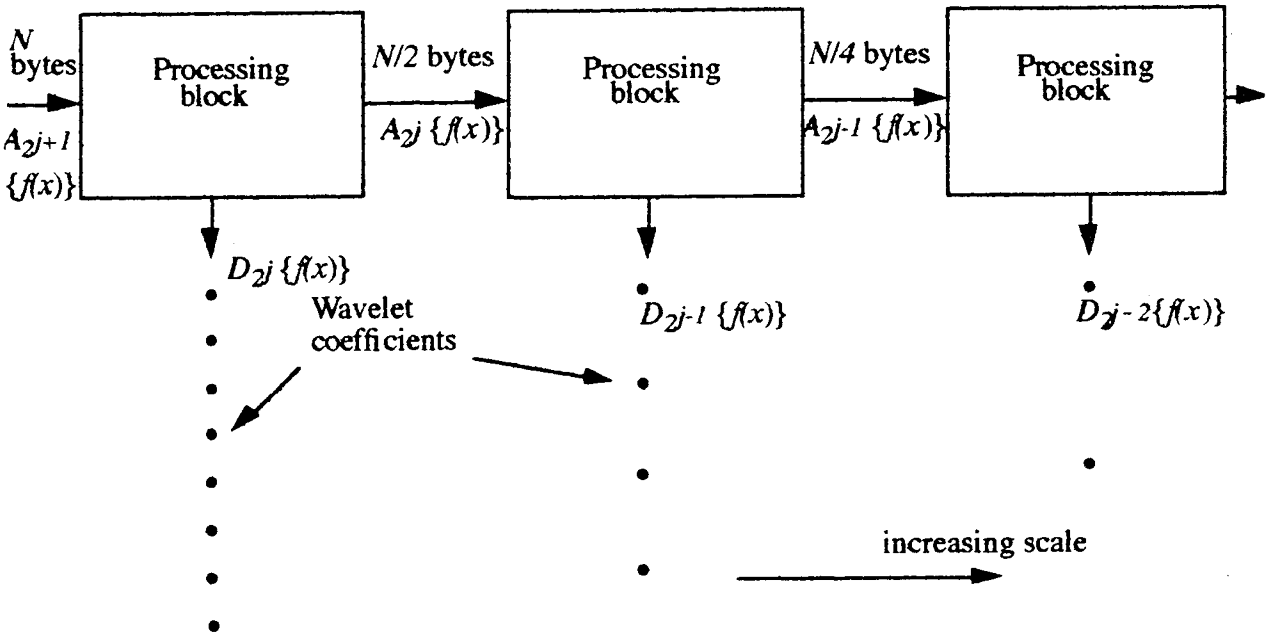


FIGURE 3. DWT decomposition

In each processing block the input is split and passed through a low-pass and high-pass filter. The impulse response of the high-pass filter $h(n)$ is the mother wavelet. The impulse response of the low-pass filter $g(n)$ is a related function called a scaling function. These filters have mirror-image symmetry about the frequency $\omega = \pi/2$ in the frequency domain and are referred to as quadrature mirror filters (QMFs).[5] If the low-pass filter has frequency response $G(\omega)$, then the high-pass filter has frequency response

$$H(\omega) = G(\omega - \pi) . \tag{7}$$

In the time domain, this can be described by

$$h(n) = (-1)^n g(n) , \tag{8}$$

where $g(n)$ and $h(n)$ are the impulse responses of the low- and high-pass filters respectively. The general relationship between the frequency responses of the scaling and wavelet functions is shown in Fig. 4.

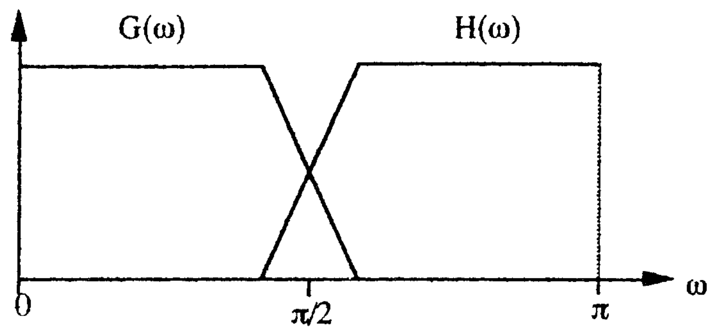


FIGURE 4. Filter characteristics of scaling function and wavelet in frequency domain

The output of each filters in a processing stage is downsampled by a factor of two. The output from the high-pass filter portion, $D_{2^j}\{f(x)\}$ are wavelet coefficients at that scale. The output from the low-pass filter portion, $A_{2^j}\{f(x)\}$ are sent to the next processing stage or scale. The schematic diagram of a processing block is shown in Fig. 5.

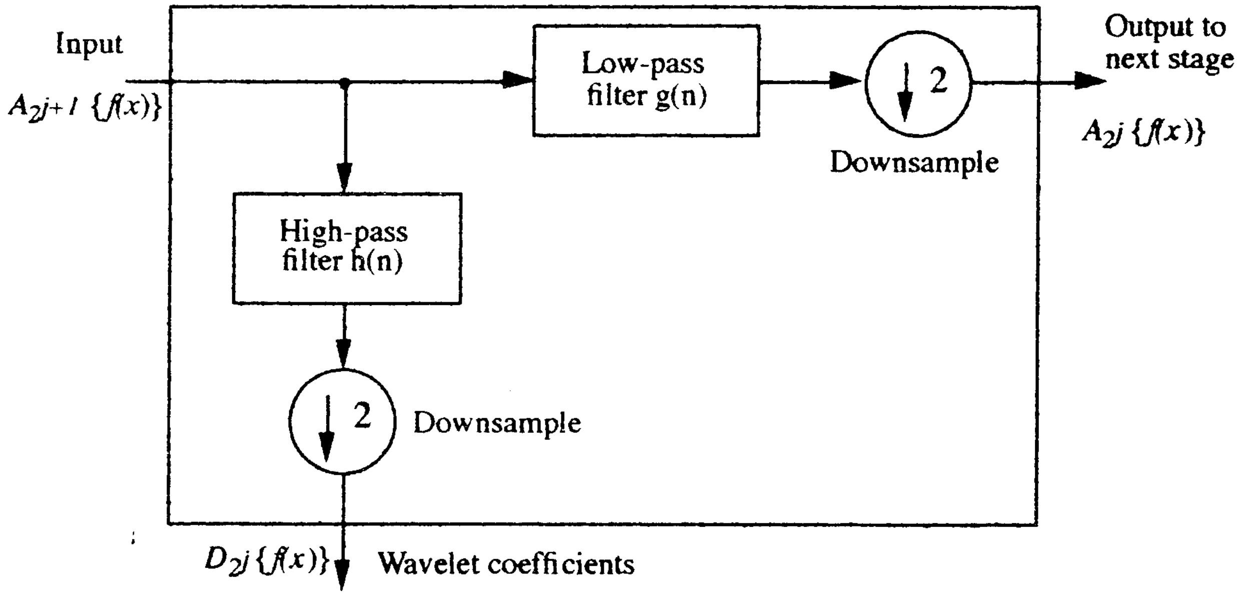


FIGURE 5. Processing block for discrete WT

2.5 Implementation of inverse WT with quadrature mirror filters

As in the discrete WT, the inverse WT can be implemented in an efficient manner by passing a signal through identically structured processing stages where each successive stage processes twice the number of bits as the previous stage. A schematic diagram of the inverse discrete WT reconstruction is shown in Fig. 6. In general N - 1 processing blocks are needed to reconstruct an N -bit time signal. As in the case of the discrete WT, wavelet coefficients correspond to resolution cells in Fig. 1.

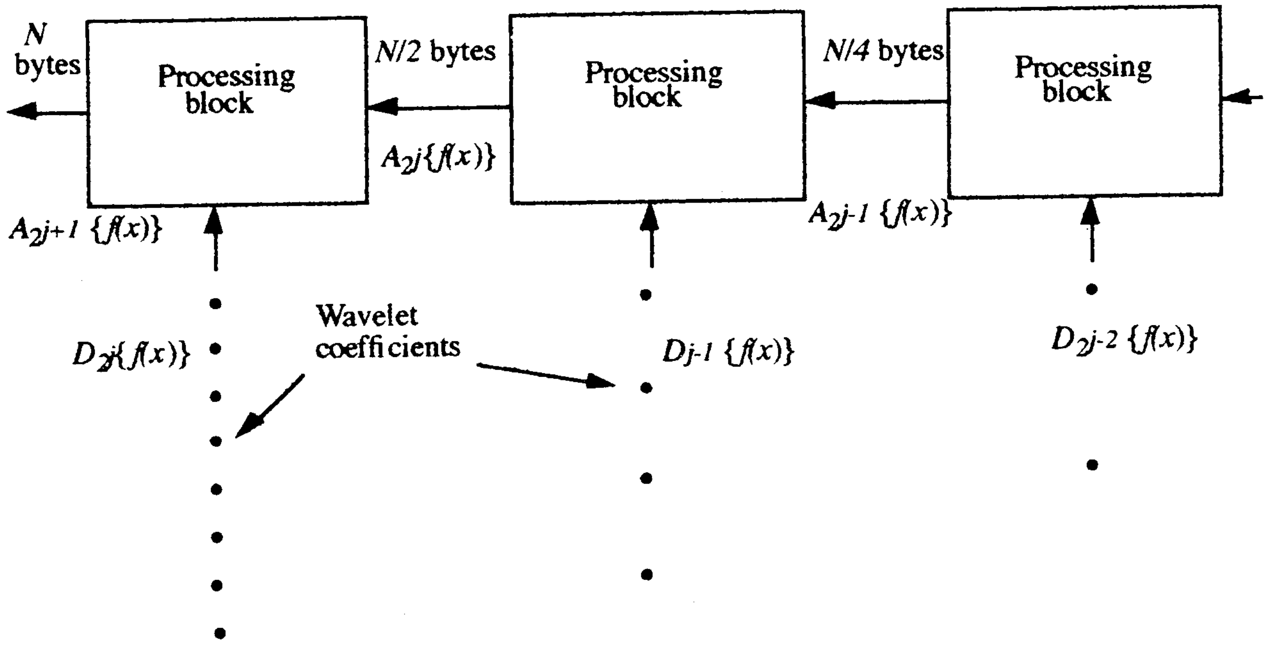


FIGURE 6. Inverse discrete WT reconstruction

In each processing block the input from the previous stage and the wavelet coefficients at that stage are upsampled with zeros then passed through separate filters. The filter processing the wavelet coefficients has impulse response $`h(n)$, and the filter corresponding to samples from the previous stage has impulse response $`g(n)$. The output of the filters are then summed together. In orthogonal transforms the decomposition and reconstruction filters in Figs. 5 and 7 respectively are the same.The schematic diagram of a processing block is shown in Fig. 7.

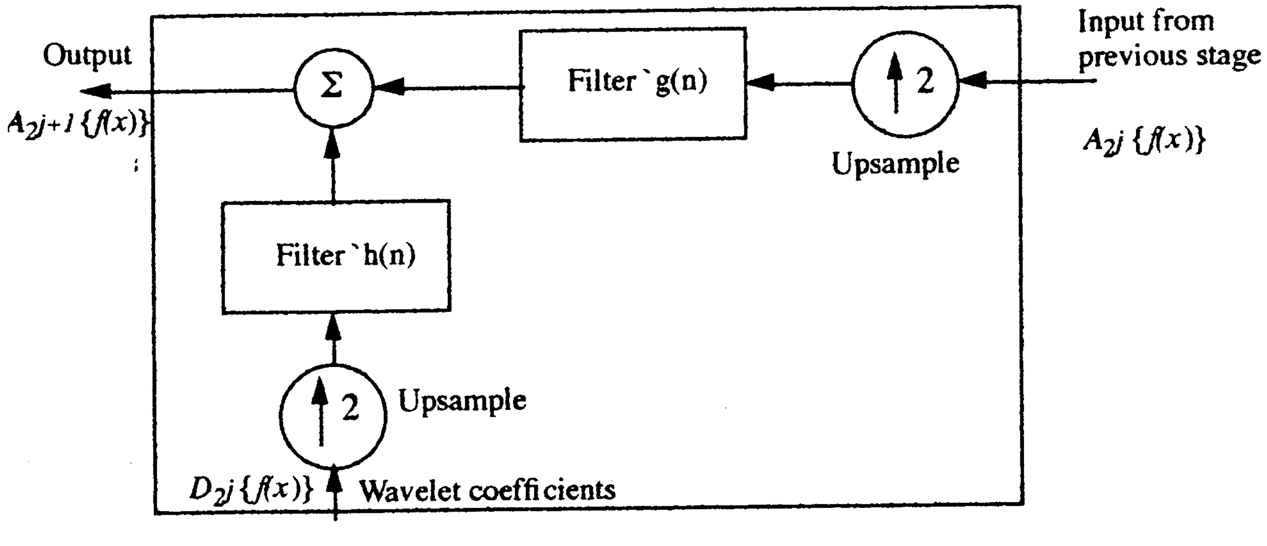


FIGURE 7. Inverse discrete WT processing block

One of the biorthogonality constraints demands that the input signal be perfectly reconstructed. Aliasing resulting from the downsampling in the discrete WT can be perfectly cancelled by properly choosing the filter coefficients in the inverse discrete WT.[5] In the time domain, this can be described by

$$\begin{aligned} {}'h(n) &= 2g(n) \\ {}'h(n) &= -2(-1)^{n}{}'g(n) \end{aligned} \tag{9}$$

where $g'(n)$ and $h'(n)$ are the impulse responses of the low- and high-pass filters respectively. In the frequency domain they can be described as

$$\begin{aligned} {}'G(\omega) &= 2G(\omega) \\ {}'H(\omega) &= -2{}'G(\omega-\pi) \end{aligned} . \tag{10}$$

The second biorthogonality constraint requires that each channel have independent information which can be written as

$$G(\omega){}'H(\omega) = 0 \; and \; H(\omega){}'G(\omega) = 0 \tag{11}$$

These conditions are imposed directly on the filter coefficients.

2.6 Examples of wavelets

Many functions exist that can serve as wavelets; however, not all of them may be of interest. We primarily considered some wavelets discovered by Daubechies.[6] They have been well-studied and shown to be useful. The wavelets we examined could be described by filter coefficients used in the filters of the previous section. Specifically, a filter is described by the following difference equation

$$y[n] = c[1]x[n] + c[2]x[n-1] + \ldots + c[M]x[n-M+1] \tag{12}$$

where $\boldsymbol{x}$ is the data vector being operated on, c's are the filter coefficients, and M is the number of coefficients in the filter. We considered Daubechies wavelets that had 4, 6, 12, and 20 filter coefficients that are tabulated in the appendix. These wavelets are referred to as Daub4, Daub6 etc.

We also considered the Haar wavelet, and one discovered by Lemaire. The Haar wavelet is most easily described in the input domain as

$$\psi(x) = \begin{cases} 1 & if\ 0 \le x < \frac{1}{2} \\ -1 & if\ \frac{1}{2} \le x < 1 \\ 0 & otherwise \end{cases} , \qquad (13)$$

its filter functions are both described in the appendix. Lemaire's wavelet is most easily described in the frequency domain by its scaling function as

$$G(\omega) = \left[2(1-u)^4 \frac{315 - 420u + 126u^2 - 4u^3}{315 - 420v + 126v^2 - 4v^3} \right]^{1/2} \qquad (14)$$

where $u = \sin^2\omega/2$, and $v = \sin^2\omega$.

We compared the scaling functions in the frequency domain of different wavelets in Fig. 8. The figure shows that the scaling function associated with the Haar wavelet has the broadest response while the scaling function associated with the Lemaire wavelet had the narrowest.

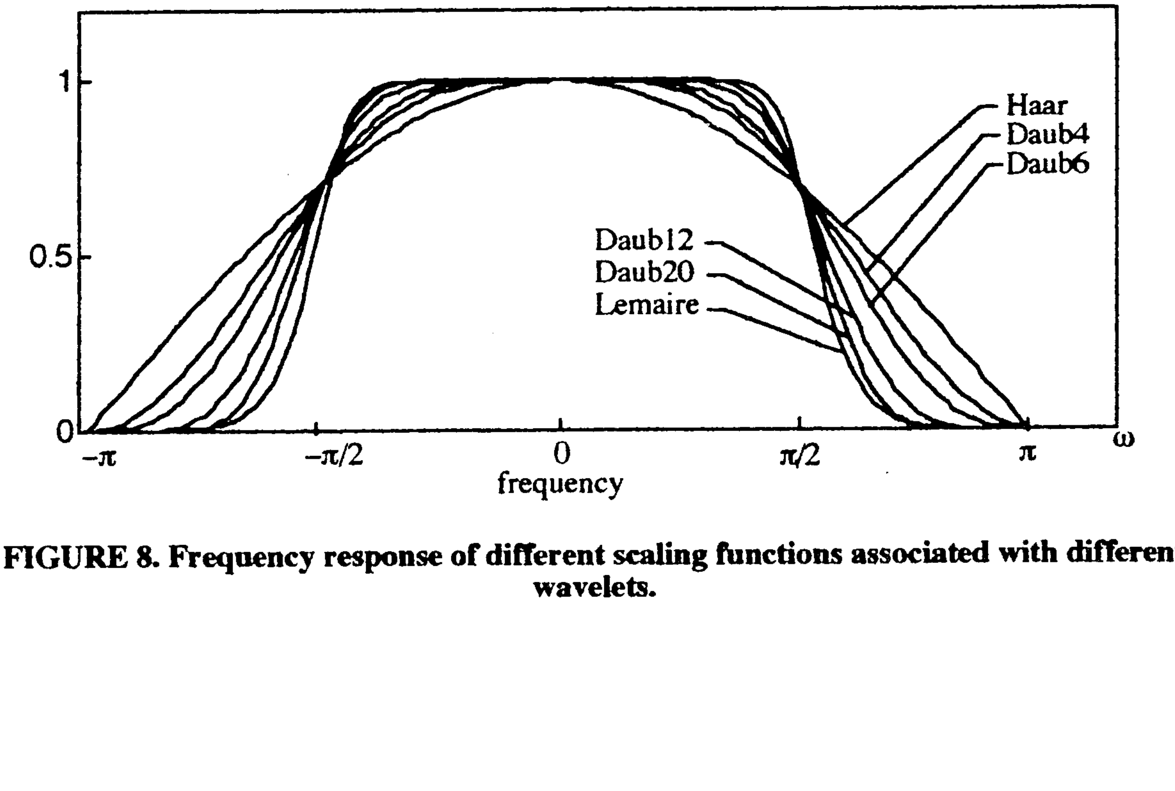


FIGURE 8. Frequency response of different scaling functions associated with different wavelets.

We also showed the relationship between the impulse response of the low- and high-pass filters specifically for the Daub4 wavelet in Fig. 9.

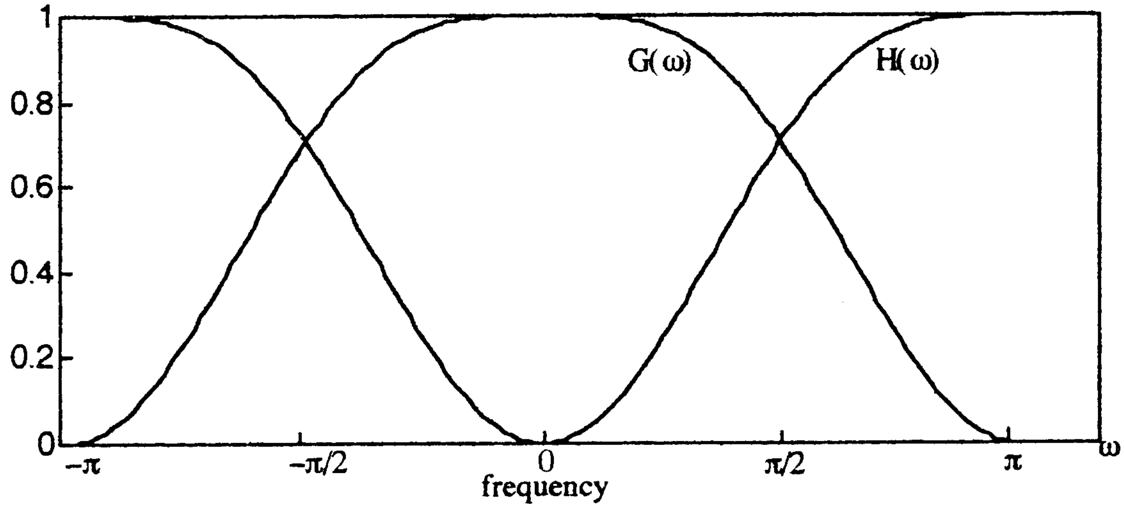


FIGURE 9. Frequency response of low- and high-pass filters (scaling function and wavelet) for Daub4 wavelet.

In Fig. 10 we showed the Daub4 wavelet in the input domain and in Fig. 11 we showed the Haar wavelet.

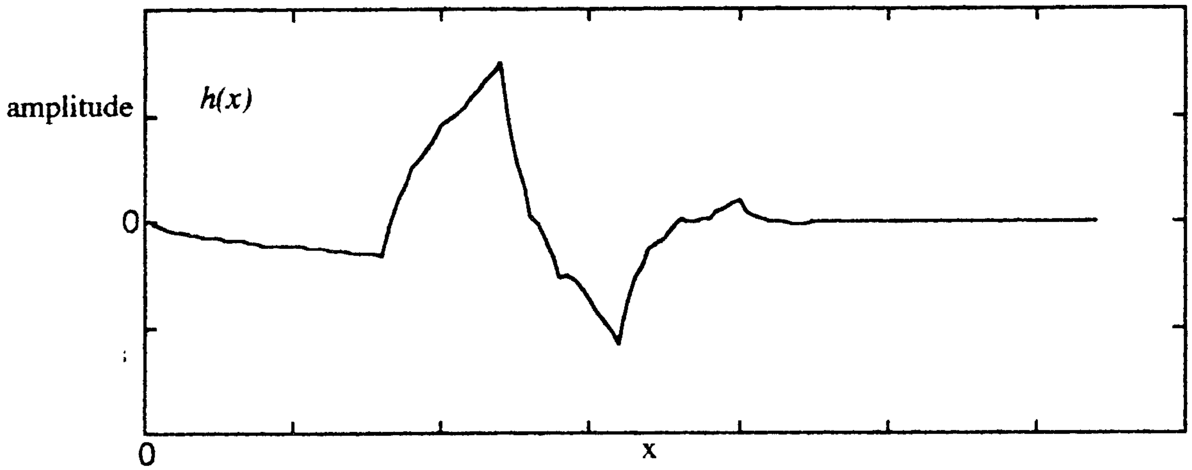


FIGURE 10. Daub4 wavelet in input domain

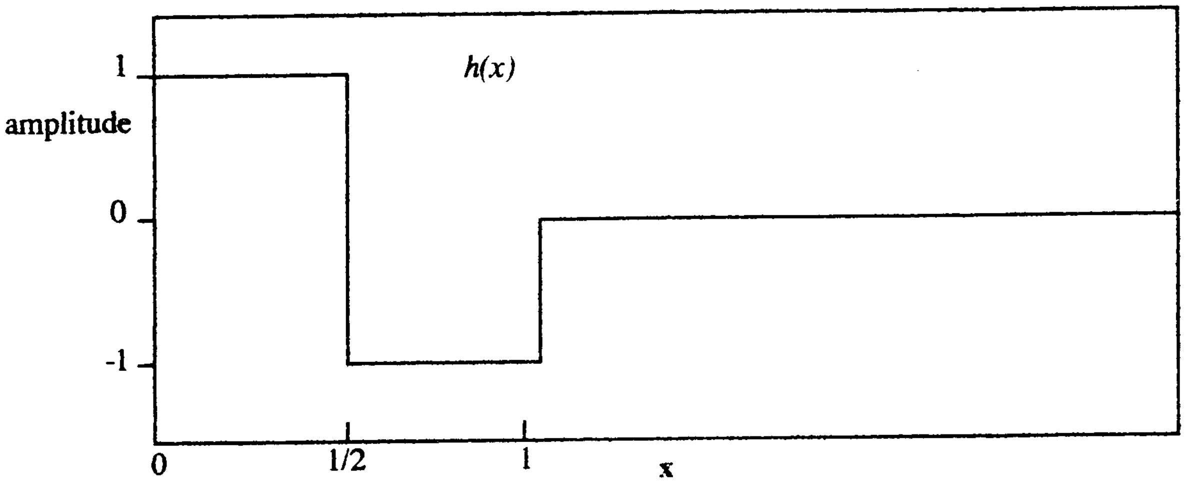


FIGURE 11. Haar wavelet in input domain

2.7 Digital implementation

The WT can be viewed as a matrix transformation operating on a vector to produce another vector in a hierarchial manner. We considered the WT described by four filter coefficients. The transformation matrix for an N-bit input where $N = 8$ looked like

$$\begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & 0 & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & 0 & 0 & 0 & c_3 & -c_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix}, \qquad (15)$$

where y is the input vector. Different size matrices used on different size input vectors would look similar with only four coefficients per row. The coefficients in the transformation matrix represent filters described in the previous section. The coefficients c_0, c_1, c_2, and c_3 correspond to a low-pass filter (scaling function) and the coefficients c_3, $-c_2$, c_1, and $-c_0$ correspond to a high-pass filter (wavelet). The same coefficients are repeated every other row shifted by two columns. The even rows perform a convolution with the low-pass filter, and the odd row perform a convolution with the high-pass filter. The matrix also downsamples the two convolutions and interleaves the two results.

The discrete WT consists of applying the matrix in Eq. (15) in a hierarchial manner. The d's in Eq. (13) are the wavelet coefficients produced at the smallest scale. Then, the s's are grouped to together to form a vector of length $N/2$ which is operated on by a $N/2$ x $N/2$ coefficient matrix. After transformation by this new matrix a new set of $N/4$ s's and $N/4$ d's are generated. These new d's are the wavelet coefficients at the next largest scale and the s's are operated on by a smaller matrix until a vector with two s's remains. The two remaining s's are called "mother-function coefficients"; however, the term "wavelet coefficients" is used to describe the two s's and all the d's. Eq. (14) shows a diagram of the calculation of the DWT on an 8-bit input vector y.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} \left\{ \left| \; transform \rightarrow \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \quad permute \rightarrow \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \right| \right\} \left\{ \left| \; transform \rightarrow \begin{bmatrix} s_1' \\ d_1' \\ s_2' \\ d_2' \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad permute \rightarrow \begin{bmatrix} s_1' \\ s_2' \\ d_1' \\ d_2' \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \right| \right\} \qquad (16)$$

The coefficients in rows 1 and 2 in the last column of Eq. (16) correspond to wavelets at the largest scale. The coefficients in rows 3 and 4 correspond to wavelets at the next largest scale, rows 5 - 8 to the next largest, and so on to rows 2^{N-1} - 2^N. The coefficients within rows corresponding to a particular scale indicate the shifts within that scale.

2.8 Inverse digital wavelet transform

To calculate the inverse discrete WT, the procedure in Eq. (16) is reversed using the inverse of the matrix in Eq. (15). In our example the inverse is the transposed matrix and the transformation looks like

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} c_0 & c_3 & 0 & 0 & 0 & 0 & c_2 & c_1 \\ c_1 & -c_2 & 0 & 0 & 0 & 0 & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 \\ c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 \\ 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & c_3 \\ 0 & 0 & 0 & 0 & c_3 & -c_0 & c_1 & -c_2 \end{bmatrix} \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix}. \tag{17}$$

As before, different size matrices would look similar with four coefficients per row. The same coefficients are repeated every other row shifted by two columns.

The inverse discrete WT consists of applying the matrix in Eq. (17) in a hierarchial manner. The *d*'s in Eq. (17) are the wavelet coefficients at a particular scale and the *s*'s are the input from the previous stage. The *s*'s and the *d*'s are contained in *N*/2-length vectors interleaved to produce an *N* -length vector. This vector is operated on by the *N* x *N* coefficient matrix. After transformation by the matrix a new set of wavelet coefficients are interleaved with the result and the process is repeated. Eq. (18) shows a diagram of the calculation of the inverse discrete WT for an 8-bit vector.

$$\begin{bmatrix} s_1' \\ s_2' \\ d_1' \\ d_2' \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \left\{ permute \rightarrow \begin{bmatrix} s_1' \\ d_1' \\ s_2' \\ d_2' \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad transform \rightarrow \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \right\} \left\{ permute \rightarrow \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \quad transform \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} \right\} \tag{18}$$

The first transformation is performed by a 4 x 4 coefficient matrix and second by an 8 x 8 coefficient matrix. The last column vector ***y*** in Eq. (18) is the reconstructed time signal.

Finally, the WT can be more computationally efficient than the FT. For an *N*-length FFT, $O(N \log_2 N)$ operations are needed. Using the filter structure in the previous section as the fast WT,

$O(N \log_2 L)$ operations are needed where L is the length of the digital filters. Therefore, the fast WT is faster than the FFT by a factor of $\log_2 N/\log_2 L$.

2.9 Two-dimensional wavelet transform

The WT can also be extended to higher dimensions. We considered the two-dimensional implementation for image processing. Eq. (4) was modified as

$$W_h f(m, n, s, t) = a_0^{\frac{-s}{2}} a_0^{\frac{-m}{2}} \iint f(x, y)\, h\,(a_0^{-m}x - nb_0, a_0^{-m}y - tb_0)\, dxdy, \qquad (19)$$

where now the WT can be considered 4-dimensional. In an analogous manner the inverse WT in two dimensions was written as

$$f(x, y) = \sum_m \sum_n \sum_s \sum_t [W_h f(m, n, s, t)]\, a_0^{\frac{-m}{2}} a_0^{\frac{-s}{2}} h\,(a_0^{-m}x - nb_0, a_0^{-m}y - tb_0). \qquad (20)$$

The two-dimensional WT can be implemented with one-dimensional convolutions of the rows and columns on an image with one-dimensional QMFs.[3] For example, the rows of an image are filtered as in the one-dimensional case. Only every other row is retained and the columns of the remaining signals are filtered and every other column retained. A block diagram of the structure is shown in Fig. 12. The image $A_{2j+1}\{f(x,y)\}$ is decomposed into four images, a lower resolution version of itself $A_{2j}\{f(x,y)\}$, and the detail images, $D^1{}_{2j}\{f(x,y)\}$, $D^2{}_{2j}\{f(x,y)\}$, and $D^3{}_{2j}\{f(x,y)\}$.

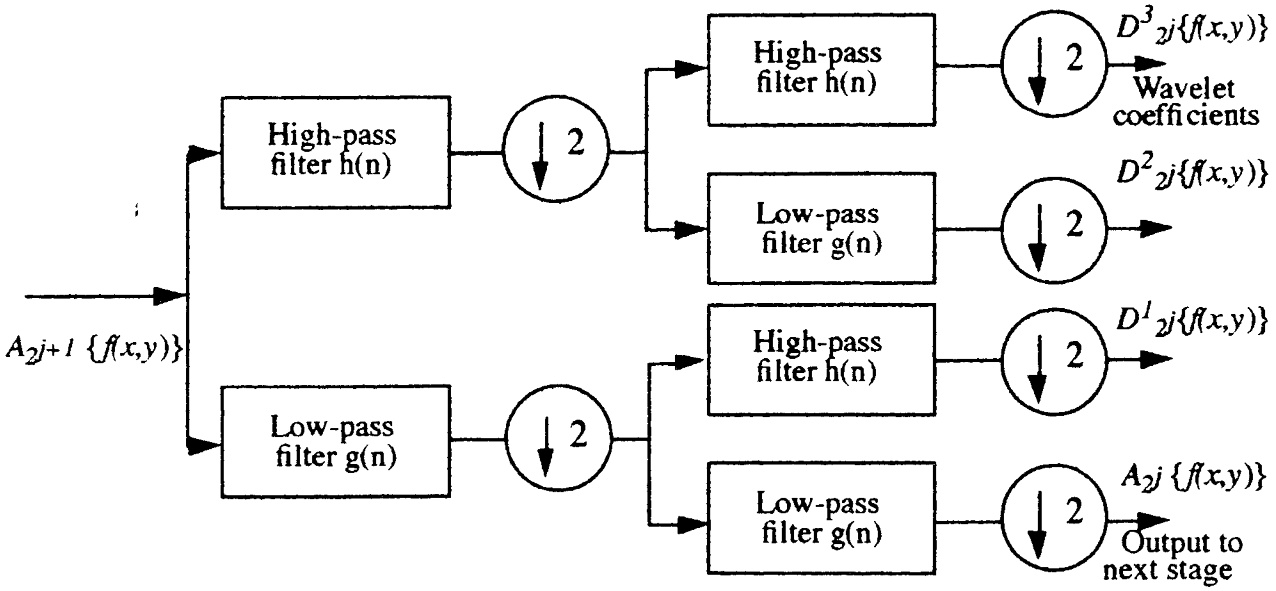
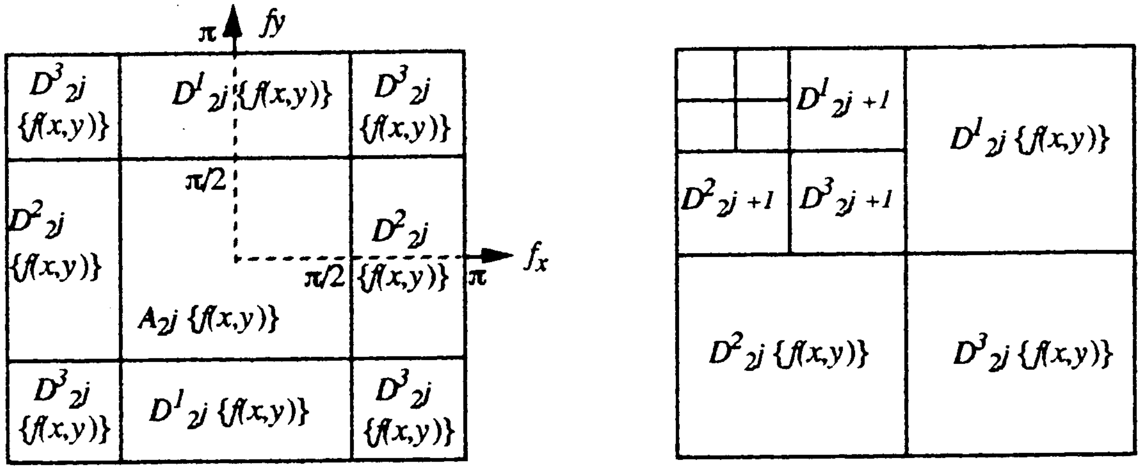


FIGURE 12. Decomposition of an image $A_{2j+1}\{f(x,y)\}$ using one-dimensional filters

The wavelet decomposition in two-dimensions can be thought of as a set of spatially oriented frequency channels. When the image $A_{2^{j+1}}\{f(x,y)\}$ is decomposed the output at a particular scale generally corresponds to regions in the Fourier plane.[3] The image $A_{2^j}\{f(x,y)\}$ corresponds to the lowest frequencies, $D^1_{2^{j+1}}\{f(x,y)\}$ corresponds to the vertical high frequencies and the horizontal low frequencies (horizontal edges), $D^2_{2^{j+1}}\{f(x,y)\}$ corresponds to the vertical low frequencies and the horizontal high frequencies (vertical edges), and $D^3_{2^{j+1}}\{f(x,y)\}$ corresponds to the vertical high frequencies and the horizontal high frequencies corners). Figure 13(a) shows the frequency support of the images which can be thought of as two-dimensional extension of Fig. 1. In addition Fig. 13(b) shows how a the wavelet representation of an image is often displayed.



(a)

(b)

FIGURE 13. Wavelet representation of an image from Ref. 3 (a) frequency support of decomposed image (b) wavelet representation of an image

In an analogous manner to wavelet decomposition, the one-dimensional reconstruction algorithm has been extended to two-dimensions.[3] At each scale, an image $A_{2^{j+1}}\{f(x,y)\}$ is reconstructed with the lower resolution version $A_{2^j}\{f(x,y)\}$ and the wavelet coefficients at that particular scale. Between each column of the images we add a column of zeros, convolve the rows with a 1-D filter, add a row of zeros between each row of the resulting image, and convolve the columns with another 1-D filter. This process is shown in Fig. 14.

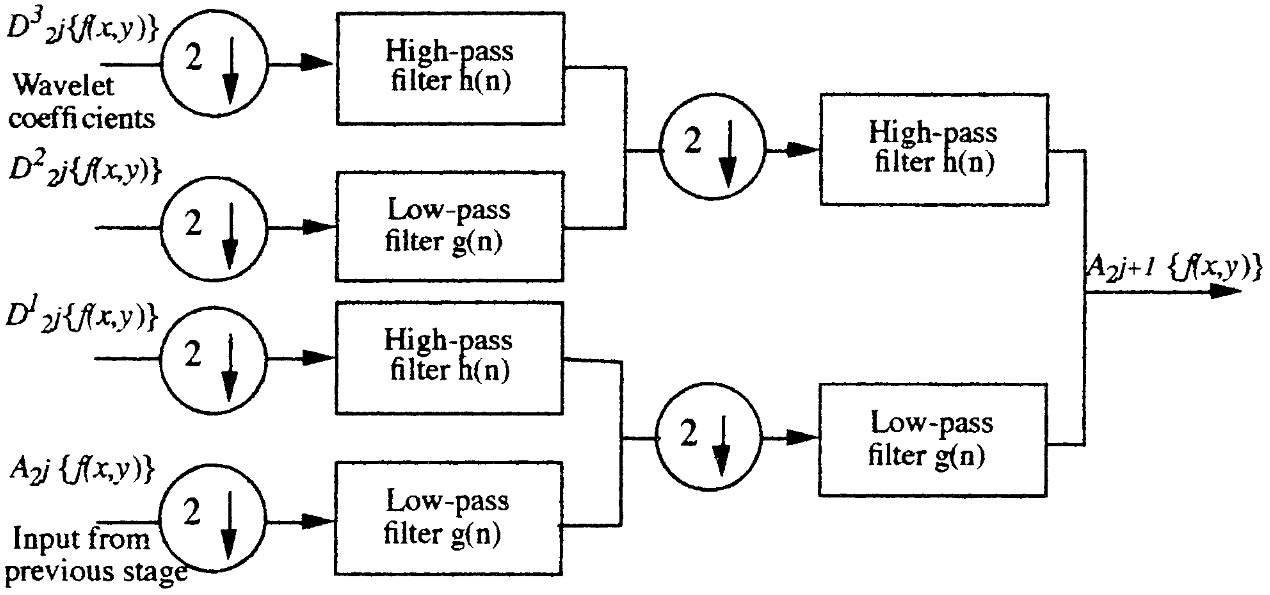


FIGURE 14. Reconstruction of the image A_{2j+1} {f(x,y) using one-dimensional filters

3.0 Optical wavelet transform

Optical processing takes advantage of the interpretation of a WT as the correlation between an input signal and a wavelet. Therefore, an optical correlator may be used to generate the WT. A wavelet at a particular scale is correlated with an image to generate all the wavelet coefficients at that scale. One optical correlation is needed for each scale of the mother wavelet.

Several experiments have been performed that use inputs recorded on film.[7-12] In most cases these have been for one-dimensional signals and both the WT and its inverse have been demonstrated. Two-dimensional systems have also been described.[13]

The Haar wavelet has been most popular with optical implementations because its the simplest wavelet and can often be implemented easily. The Haar wavelet can be implemented with a binary spatial light modulator (SLM) that has a zero state.[14] Using this approach the Haar wavelet was placed in the input plane of a 4f optical correlator, and a hologram of an image was placed in the Fourier plane. A family of Haar wavelets were correlated with the input image to generate the magnitude of the WT.[14] One drawback was that the binary device could not produce the scaling factor associated with a wavelet. Furthermore, the binary SLM had difficulty producing a true zero state required for the family of wavelets which degraded the results. Finally, the results were degraded by a DC component present in the wavelet's frequency domain.

Another approach chose a wavelet with a positive real binary FT, then approximated the wavelet with a positive real binary pattern.[15] A joint-transform correlator based on the WT has been also demonstrated.[16] In this work, the input and reference object were implemented on a pixelated device and the first five orders of the joint-FT were imaged onto a power-law device.

Then, a fixed mask containing five wavelet filters was imaged onto the power spectrum of the inputs. In this way, the output correlation response consisted of five correlations, one for each wavelet filter.

Optical processing of the WT has been applied to the processing of synthetic aperture radar (SAR).[17] In this work an acousto-optic cell was used in the input plane of an optical correlator using simulated SAR data and a liquid crystal television in the filter plane. An optical composite filter was introduced that was able to yield desired outputs for a given set of inputs.[18] There are several other interesting papers in Refs. 19-21 including algorithms used for the detection of objects.[22]

We described a one-dimensional set-up for implementing the WT and its inverse optically using four scales. This configuration was used in Ref. 15 and is basically two cascaded optical correlators. A schematic diagram of the system is shown in Fig. 15 with the distance between a lens and a plane equal to f, the focal length of the lens. A cylindrical lens first focuses collimated light on a one-dimensional input in plane P1. The spherical lens then produces the FT of the input in plane P2. The FTs will spread in the y-direction. Therefore, placing a bank of FTs of wavelets at different scales oriented in the x-direction will produce correlations between the input and the wavelets in parallel. In this diagram the scale increases as a increases. The resulting correlation between the input signal and the wavelets will appear in plane P3. Inserting a bank of wavelets in plane P3 produces the WT in plane P4, the convolution between the WT and the wavelets. Orthogonal views of Fig. 15 are shown in Fig. 16.

Because the FT can be implemented relatively efficiently with optics, an optical correlator has been shown to be useful for object recognition. Because the WT and its inverse can be viewed as correlation operations, it seems that signal processing using wavelets could benefit from an optical implementation. However, some important considerations must be acknowledged. The first is the increase in dimensionality of the processing. When viewed as correlation operations, the WT increases the dimensionality of the system. For 1-D systems a 2-D wavelet domain must be processed. For the 2-D case, a 4-D or multiple 2-D domains must be processed.

As the scale of the wavelet increases fewer number of coefficients are needed and the optical implementation becomes less efficient. For a one-dimensional input, only half the number of pixels as the input are needed to determine the wavelet coefficients at the smallest scale. At the next largest scale only 1/4 the pixels are needed. At the next two larger scales only 1/8 and 1/16 the number of pixels are needed respectively. Generally an optical system has a fixed resolution so that the optical system must be designed for use with the highest resolution system even though the resolution is not always needed. For example, to perform the WT in parallel for a N-byte one-dimensional input, a j x N pixel wavelet domain must be considered where j is the number of scales considered. For the two-dimensional case, a j^2N^2 wavelet domain must be considered for an N x N input image. Due to the limited resolution of existing SLMs, this can be a serious limitation for optical wavelet processing of images.

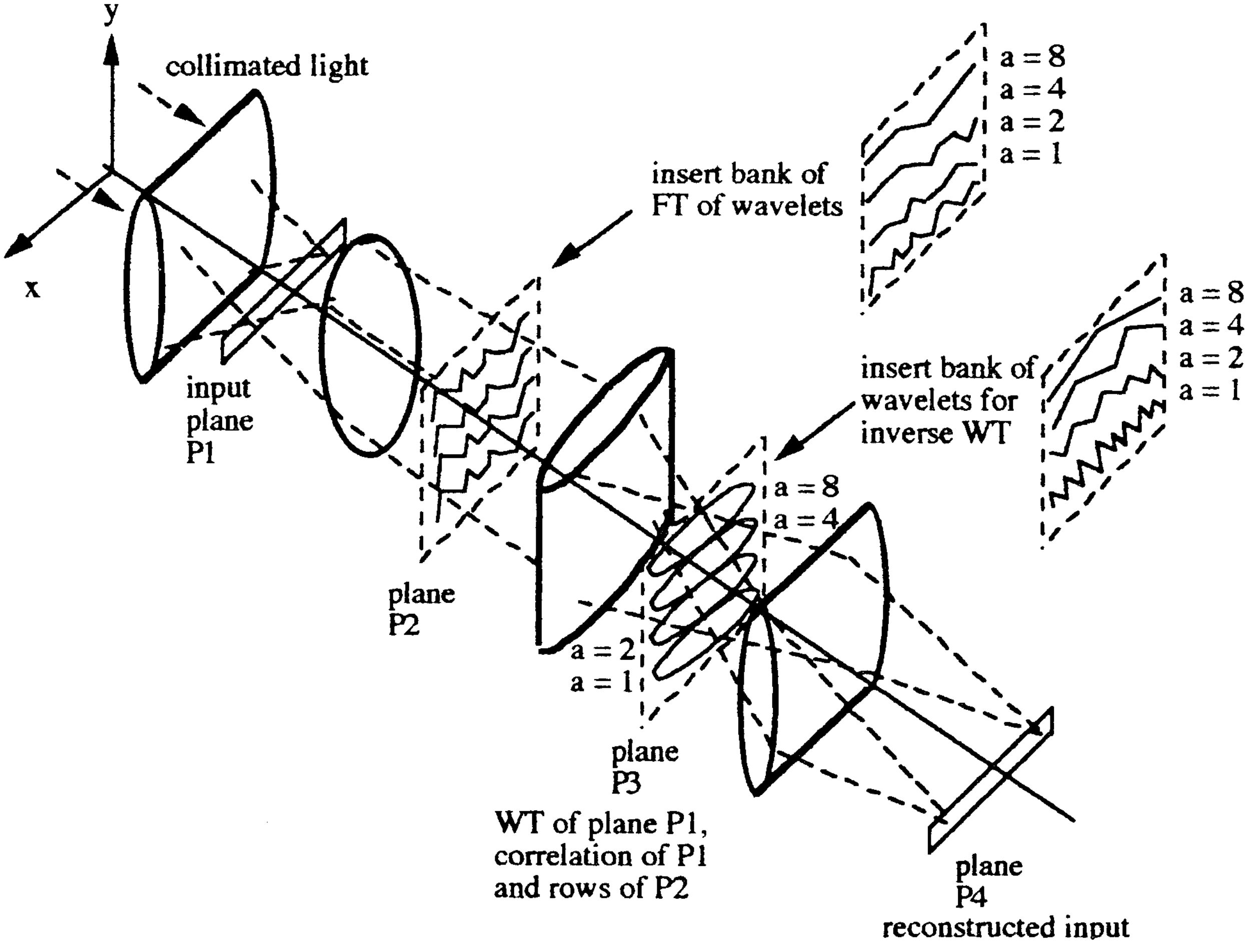


FIGURE 15. Optical implementation of the WT and inverse WT for a one-dimensional input. Distances between lenses and planes are equal to *f*.

Finally, the negative numbers generated or needed by the WT and its inverse may be difficult to implement optically. As in optical FT processing, complex numbers can be handled by performing operations using coherent light. Such an approach requires the full resolution of each scale in the WT. In other words, the WT or its inverse cannot generally be performed sequentially with a lower resolution optical system. In spite of the drawbacks of implementing the WT optically, there may be areas where optics may be useful. Perhaps in examining a limited number of scales of a WT or multiresolution processing where the above considerations can be minimized or eliminated.

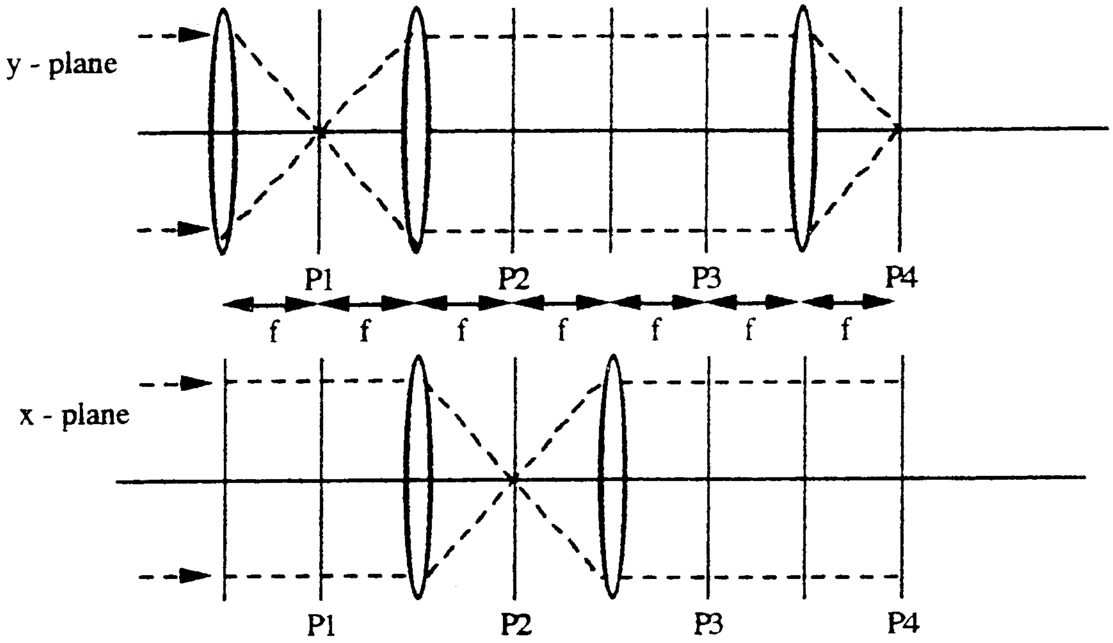


FIGURE 16. Orthogonal views of Fig. 16, *y*-plane is the side view and the *x*-plane is the top view.

4.0 Multiresolution processing

In multiresolution processing, a signal is processed multiple times, each time at a different resolution. An image is generally represented by a series of lower-resolution versions of itself, including the original image. In the case of identifying objects, a low-resolution version of an image could be used to find regions of interest in an image that gives only the approximate size and location of an object. To determine more precisely the size and location of an object, information must be passed down from the lower-resolution version of the image to a higher-resolution version. This procedure is repeated until the highest resolution of the pyramid is reached.

A limiting factor in the application of optical correlators is that the number of pixels of currently available spatial light modulators (SLMs) is often not large enough for a variety of applications. Many applications require at least 512 x 512 pixels, which is larger than most SLMs, and applications requiring more pixels are not uncommon. One approach to this problem is to process large images at lower resolutions beginning with the lowest resolution. In this way, knowledge about an object's shape or location is made more precise on successively higher resolution versions of an image.

Pyramidal processing is a form of multi-resolution processing, where an image is processed multiple times, each time at a different resolution.[23] In pyramid processing, an image is represented by a series of lower-resolution versions of itself, including the original image. Using digital image processing techniques, template matching has been previously investigated using a pyramid structure to identify objects or their features.[24] For example, a low-resolution version of an image was used to find regions of interest in an image that give only the approximate size and location of an object. To determine more precisely the size and location of an object, information must be passed down from the lower-resolution version of the image to a higher-resolution version. This procedure is repeated until the highest resolution of the pyramid is reached. An optical technique may be used to generate layers of the pyramid,[25] or an image can be processed with a lower-resolution optical correlator for object recognition using pyramid processing.[26]

4.1 Multiresolution processing using the wavelet representation

The data in the pyramid structures are correlated. As pointed out in Ref. 3 it is not clear whether a similarity between the image details at different resolutions is due to a property of the image itself or to the redundancy of the representation. In the orthogonal wavelet representation there is no oversampling and data at different levels are independent. The wavelet representation allows a multiresolution representation to be constructed based on the difference of information available at two successive resolutions.[3]

We have previously introduced a multiresolution operator A_{2^j} in Sec. 2.3. Here, we expand on this concept and consider some of the properties of multiresolution processing.[3] The operator A_{2^j} is an orthogonal projection on a vector space, and $A_{2^j}\{f(x)\}$ is not modified if we approximate it again at resolution 2^j. In addition, $A_{2^j}\{f(x)\}$ is the function which is most similar to $f(x)$ at resolution 2^j. Furthermore, the approximation of a signal at a resolution 2^j contains all the necessary information to compute the signal at resolution 2^{j-1}. Finally, the approximation operation is similar at all resolutions.

The projection of $f(x)$ onto a vector space can be achieved with a convolution operation. The approximation of a signal $f(x)$ at resolution 2^j can be viewed as a convolution between the signal $f(x)$ and a scaling function $\phi(x)$ followed by a uniform sampling at the rate of 2^j. The approximation can be written as

$$A_{2^j}\{f(x)\} = \int f(u)\,\phi_{2^j}(u - 2^{-j}n)\,du. \tag{21}$$

In two dimensions Eq. 21 can be written as

$$A_{2^j}\{f(x,y)\} = \iint f(u,v)\,\phi_{2^j}(u - 2^{-j}n)\,\phi_{2^j}(v - 2^{-j}m)\,dudv\ , \tag{22}$$

if a separable scaling function is used. As indicated in Sec. 2.3 the signal $A_1\{f(x)\} = f(x)$, the original signal at the highest resolution. The function $f(x)$ at the resolution 2^j can be referred to as $A_{2^j}\{f(x)\}$ with $j < 0$. The signals $A_{1/2}\{f(x)\}$, $A_{1/4}\{f(x)\}$, etc. are lower resolution versions of $f(x)$.

The multiresolution approximation is completely characterized by the scaling function. It is possible to chose scaling functions with good localization properties in both the frequency and input domains.

The detail signal $D_{2^j}\{f(x)\}$ at a resolution 2^j contains the difference of information between two successive approximations $A_{2^{j+1}}\{f(x)\}$ and $A_{2^j}\{f(x)\}$. The detail signal at resolution 2^j can be viewed as a convolution between the signal $f(x)$ and the wavelet $h(x)$ followed by a uniform sampling at the rate of 2^j. The detail signal can be written as

$$D_{2^j}\{f(x)\} = \int f(u)\, h_{2^j}(u - 2^{-j}n)\, du, \tag{23}$$

where the wavelet $h(x)$ can be generally thought of as a bandpass filter in the frequency bands $|[2^{-j}\pi/2, 2^{-j+1}\pi]|$. In two dimensions there are three detail signals corresponding to the separable products of $h(x)$ and $\phi(x)$. The detail signals can be written in two dimensions as

$$D^1_{2^j}\{f(x,y)\} = \iint f(u,v)\, \phi_{2^j}(u - 2^{-j}n)\, h_{2^j}(v - 2^{-j}m)\, dudv \tag{24}$$

$$D^2_{2^j}\{f(x,y)\} = \iint f(u,v)\, h_{2^j}(u - 2^{-j}n)\, \phi_{2^j}(v - 2^{-j}m)\, dudv \tag{25}$$

$$D^3_{2^j}\{f(x,y)\} = \iint f(u,v)\, h_{2^j}(u - 2^{-j}n)\, h_{2^j}(v - 2^{-j}m)\, dudv \tag{26}$$

where their frequency support was illustrated in Fig. 13(a).

For the one-dimensional case, an orthogonal representation of a signal consists of the original signal at a coarse resolution $-J$ and the detail signals at higher resolutions $-J \le j \le -1$. The representation can be written as

$$(A_{2^{-J}}\{f(x)\}, D_{2^j}\{f(x,y)\}_{-J \le j \le -1}) \, . \tag{27}$$

where 2^{-J} indicates the coarsest resolution. In two dimensions an image is completely represented by that image at a coarse resolution and the $3J + 1$ detail images as

$$(A_{2^{-J}}\{f(x,y)\}, D^1_{2^j}\{f(x,y)\}_{-J \le j \le -1}, D^2_{2^j}\{f(x,y)\}_{-J \le j \le -1}, D^3_{2^j}\{f(x,y)\}_{-J \le j \le -1}) \tag{28}$$

The number of pixels in this representation is equal to the number of pixels in the original image which is due to the orthogonality of the representation. An illustration of the decomposition is represented in Fig. 13(b).

4.2 Optical multiresolution processing

Because each level of a multiresolution representation represents a different resolution, each level contains a different size image. Because we usually process only a constant size image with an optical correlator, we processed only a portion of an image when the resolution is increased. We considered a hybrid approach to multiresolution processing by using a computer to filter and downsample an image, then used an optical system to perform correlations between the image and a reference. Using this technique large images can be searched in parallel using an optical correlator.[26] We compared the performance of multiresolution processing using the scaling functions described previously to the optimal result. We also showed by simulation the results of processing a 256 x 256 pixel image from an infrared sensor with a 128 x 128 pixel optical correlator.

We examined the performance by computer simulation of a binary optical correlator using autocorrelation experiments. We compared the performance of the scaling functions to an ideal low-pass filter because the ideal low-pass filter provides the optimum result. We used a 256 x 256 image consisting of an object in a constant background in our autocorrelation experiments. The central 128 x 128 pixels are shown in Fig. 17. We performed digital filtering on the 256 x 256 image using the different scaling functions followed by downsampling to create a 128 x128 image. To form the reference filter, the 256 x 256 image was ideally low-pass filtered, downsampled, then made into a binary phase-only filter (BPOF). Correlations were then performed between the differently-filtered input image and the BPOF. The signal-to-noise ratio (SNR) and the peak-to-correlation energy (PCE) were used as performance measures. The PCE can be considered as a measure of correlation peak sharpness.

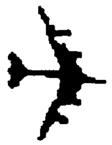


FIGURE 17. Image used in autocorrelation experiments

The SNR and PCE combination that corresponds to a each scaling function are tabulated in Table 1. The SNR and PCE resulting from an ideally low-pass filtered input image were each set to 100 for comparison. It can be seen that the choice of scaling function significantly affects the SNR and PCE of the autocorrelation result. The scaling function associated with the Lemaire wavelet had almost the same performance as the ideal low-pass filter. For all scaling functions both the SNR and PCE were lower than the ideal low-pass filter result.

There does not seem to be an obvious pattern to the performance of the different scaling functions. It appears that perhaps the Lemarie scaling function produced the overall best result. However, the scaling function most similar to the Lemaire scaling function, the Daub20 gave the poorest performance. The Haar scaling function, which is the least similar to the ideal low-pass filter produced results better than half the scaling functions considered.

TABLE 1. Performance using different scaling operations

Scaling operation	SNR	PCE
Ideal low-pass	100	100
Haar	88.3	78.8
Daub4	94.2	90.0
Daub6	59.4	58.9
Daub12	18.7	22.1
Daub20	15.6	16.7
Lemaire	97.1	99.4

We compared the performance of the scaling functions using the BPOF described earlier to that of the ideal low-pass filter when some frequencies of the BPOF were eliminated. It has been previously shown that the SNR can be increased by setting the filter magnitude to zero where the FT of the input image is small.[27] The region of transmitting pixels associated with a particular filter is referred to as the support region. We plotted both the SNR and the PCE as a function of transmitting pixels in Fig. 18. Each value of transmitting pixels indicates a different support function. The graph shows that there is a trade-off between SNR and PCE. Generally, as the SNR increased the PCE decreased. The values of PCE corresponding to the region where the SNR is a maximum is often, as in this case, unacceptable. Therefore, a good compromise is usually sought.

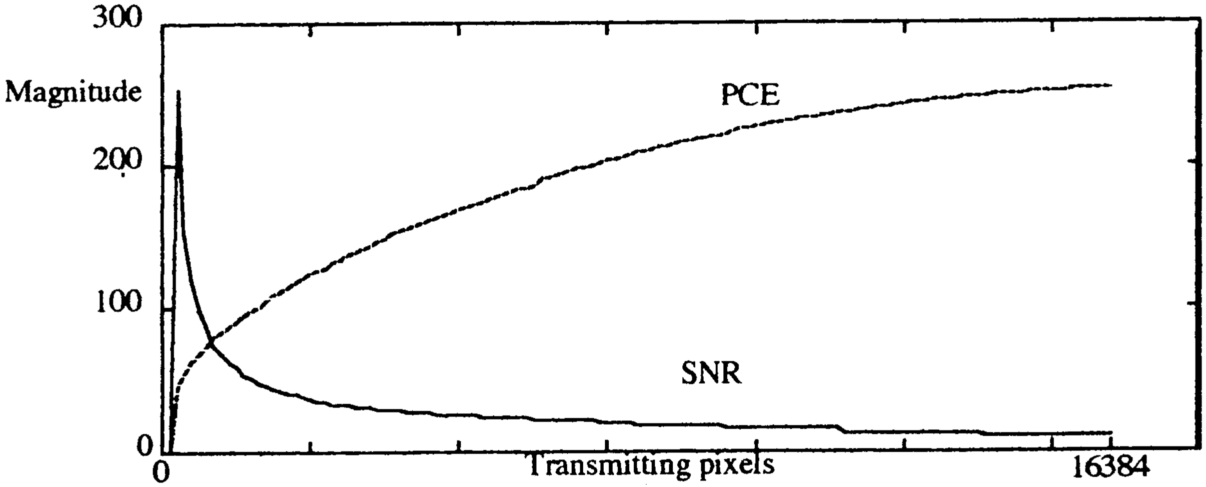


FIGURE 18. SNR and PCE for different support functions for autocorrelation of 256 x 256 ideally low-pass filtered image

The trade-off between SNR and PCE is often seen better with a graph of SNR as a function of PCE. The data in Fig. 18 was plotted in Fig. 19 to examine this trade-off. In addition, the SNR - PCE combinations associated with the scaling functions are also indicated. The results show that the performance of some scaling functions appear to be similar to the performance of some of the support functions. However, the performance of other scaling functions does not appear to be similar to some support functions.

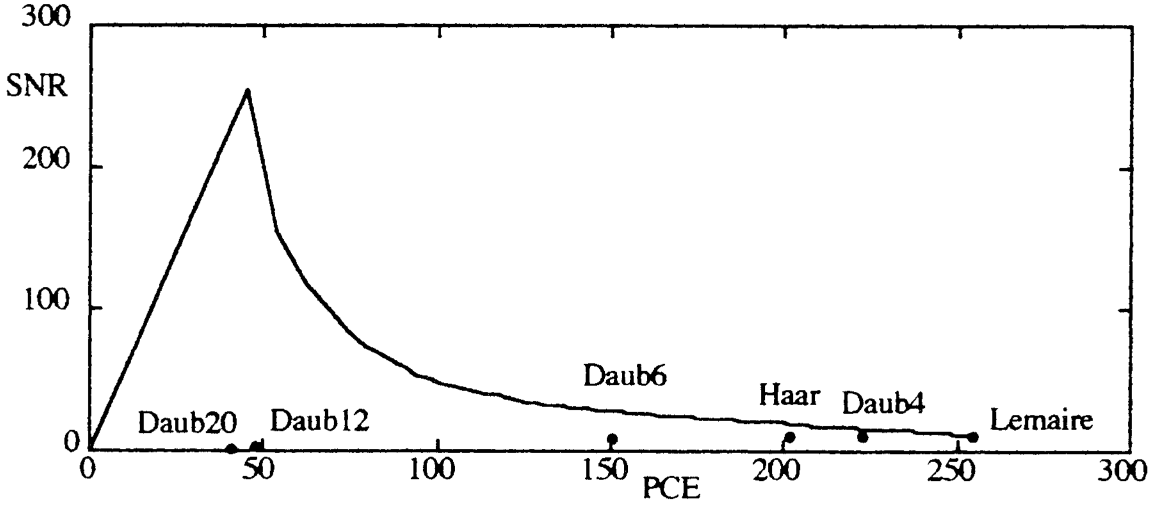


FIGURE 19. Graph of SNR as a function PCE for different support functions for autocorrelation of 256 x 256 ideally low-pass filtered image, and performance of scaling functions using all pixels of filter transmitting

5.0 Summary and Conclusion

Wavelets appear to be an interesting topic of research with many possible applications to areas such as pattern recognition, image processing, and image compression to name a few. However, the role optics is to play in terms of implementations is not clear.

Optical processing takes advantage of the interpretation of a WT as the correlation between an input signal and a wavelet. Therefore, an optical correlator may be used to generate the WT. A wavelet at a particular scale is correlated with an image to generate all the wavelet coefficients at that scale and several experiments have been demonstrated.

Some important considerations must be acknowledged for a practical implementation. The first is the increase in dimensionality of the processing. When viewed as correlation operations, the WT increases the dimensionality of the system. For 1-D systems a 2-D wavelet domain must be processed. For the 2-D case, a 4-D or multiple 2-D domains must be processed.

As the scale of the wavelet increases fewer number of coefficients are needed and the optical implementation becomes less efficient. Generally an optical system has a fixed resolution so that the optical system must be designed for use with the highest resolution system even though the resolution is not always needed. Due to the limited resolution of existing SLMs, this can be a serious limitation for optical wavelet processing of images. Finally, the negative numbers generated or needed by the WT and its inverse may be difficult to implement optically.

In spite of the drawbacks of implementing the WT optically, there may be areas where optics may be useful. Perhaps in examining a limited number of scales of a WT or multiresolution processing where the above considerations can be minimized or eliminated. Multiresolution processing would probably benefit from a hybrid approach. Digital processing could generate different resolution images and select portions of an image to be processed. Then an optical correlator could be used to process selected imagery further.

6.0 Appendix

TABLE 2. Filter coefficients for Haar, Daub4 and Daub6 wavelets

Haar		Daub4		Daub6	
Scaling	Wavelet	Scaling	Wavelet	Scaling	Wavelet
c0 = 0.70711	c0' = c1 = 0.70711	c0 = 0.48296	c0' = c3 = -0.12941	c0 = 0.3327	c0' = c5 = 0.0352
c1 = 0.70711	c1' = -c0 = -0.70711	c1 = 0.83652	c1' = -c2 = -0.22414	c1 = 0.8069	c1' = -c4 = 0.0854
		c2 = 0.22414	c2' = c1 = 0.83652	c2 = 0.4599	c2' = c3 = -0.1350
		c3 = -0.12941	c3' = -c0 = -0.48296	c3 = -0.1350	c3' = -c2 = -0.4599
				c4 = -0.0854	c4' = c1 = 0.8069
				c5 = 0.0352	c5' = -c0 = -0.3327

TABLE 3. Filter coefficients for Daub12 and Daub20 wavelets

Daub12		Daub20	
Scaling	Wavelet	Scaling	Wavelet
c0 = 0.11154	c0' = c11 = -0.0010773	c0 = 0.026670	c0' = c19 =-0.000013264
c1 = 0.49462	c1' = -c10 =-0.0047773	c1 = 0.18818	c1' =-c18 =-0.000093589
c2 = 0.75113	c2' = c9 = 0.00055384	c2 = 0.52720	c2' = c17 = -0.00011647
c3 = 0.31525	c3' = -c8 = 0.031582	c3 = 0.68846	c3' = -c16 = 0.00068586
c4 = -0.22626	c4' = c7 = 0.027523	c4 = 0.28117	c4' = c15 = 0.0019924
c5 = -0.12977	c5' = -c6 = -0.097502	c5 = -0.24985	c5' = -c14 = -0.0013954
c6 = 0.097502	c6' = c5 = -0.12977	c6 = -0.19595	c6' = c13 = -0.010733
c7 = 0.027523	c7' = -c4 = 0.22626	c7 = 0.12737	c7' = -c12 = -0.0036066
c8 = -0.031582	c8' = c3 = 0.31525	c8 = 0.093057	c8' = c11 = 0.033213
c9 = 0.00055384	c9' = -c2 = -0.75113	c9 = -0.071394	c9' = -c10 = 0.029458
c10 = 0.0047773	c10' = c1 = 0.49462	c10 = -0.029458	c10' = c9 = -0.071394
c11 = -0.0010773	c11' = -c0 = -0.11154	c11 = 0.033213	c11' = -c8 = -0.093057
		c12 = 0.0036066	c12' = c7 = 0.12737
		c13 = -0.010733	c13' = -c6 = 0.19595
		c14 = 0.0013954	c14' = c5 = -0.24985
		c15 = 0.0019924	c15' = -c4 = -0.28117
		c16 = -0.00068586	c16' = c3 = 0.68846
		c17 = -0.00011647	c17' = -c2 = -0.52720
		c18 = 0.000093589	c18' = c1 = 0.18818
		c19 = -0.000013264	c19' = -c0 = -0.026670

TABLE 4. Filter coefficients for Lemaire wavelet

Lemaire	
Scaling	Wavelet
c0 = 0.011237	c0’ = -c18 = -0.011237
c1 = 0.009490	c1’ =c17 = 0.009490
c2 = -0.02426	c2’ = -c16 = 0.02426
c3 = -0.019294	c3’ = c15 = -0.019294
c4 = 0.055723	c4’ = -c14 = -0.055723
c5 = 0.036647	c5’ = c13 = 0.036647
c6 = -0.14294	c6’ = -c12 = 0.14294
c7 = -0.057687	c7’ = c11 = -0.057687
c8 = 0.55641	c8’ = -c10 = -0.55641
c9 = 0.97299	c9’ = c9 = 0.97299
c10 = 0.55641	c10’ = -c8 = -0.55641
c11 = -0.057687	c11’ = c7 = -0.057687
c12 = -0.14294	c12’ = -c6 = 0.14294
c13 = 0.036647	c13’ = c5 = 0.036647
c14 = 0.055723	c14’ = -c4 = -0.055723
c15 = -0.019294	c15’ = c3 = -0.019294
c16 = -0.02426	c16’ = -c2 = 0.02426
c17 = 0.009490	c1’ =c1 = 0.009490
c18 = 0.011237	c18’ = -c0 = -0.011237

7.0 References

[1] Y. Meyer, “Wavelets: algorithms and applications”, SIAM: Philadelphia (1993)

[2] R. K. Young, “Wavelet theory and its applications”, Kluwer Academic Publishers: Boston (1993)

[3] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Trans. PAMI* vol. 11(7), 674-693 (1989)

[4] I. Daubechies, ed., “Different perspectives on wavelets,” American Mathematical Society: Providence, RI (1993)

[5] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, “Advanced signal processing,” Macmillan: New York (1992)

[6] I. Daubechies, “Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, vol. 41, 909-996 (1988)

[7] H. Szu, Y. Sheng, and J. Chen, “Wavelet transform as a bank of matched filters,” *Appl. Opt.* vol. 31(17), 3267-3277 (1992)

[8] E. Freysz, B. Pouligny, F. Argoul, and A. Arneodo, “Optical wavelet transform of fractal aggregates,” *Phy. Rev. Lett.* vol. 64(7), 745-748 (1990)

[9] Y. Zhang, and Y. Li, “Optical determination of Gabor coefficients of transient signals,” *Opt. Lett.* vol. 16(13), 1031-1033 (1991)

[10] Y. Zhang, Y. Li, E. G. Kanterakis, A. Katz, and X. J. Lu, “Optical realization of wavelet transform for a one-dimensional signal,” *Opt. Lett.* vol. 17(3), 210-212 (1992)

[11] Y. Li, and Y. Zhang, “Coherent optical processing of Gabor and wavelet expansions of one-dimensional and two-dimensional signals,” *Optical Engineering* vol. 31(9), 1865-1885 (1992)

[12] Y. Sheng, D. Roberge, and H. Szu, “Optical wavelet transform,” *Optical Engineering* vol. 31(9), 1840-1845 (1992)

[13] Y. Sheng, D. Roberge, and H. J. Caulfield, “Optical implementation of a two-dimensional wavelet transform,” *Optical Engineering* vol. 31(9), 1859-1864 (1992)

[14] T. J. Burns, K. H. Fielding, S. K. Rogers, S. D. Pinski, and D. Ruck, “Optical Haar wavelet transform,” *Optical Engineering* vol. 31(9), 1852-1858 (1992)

[15] A. D. McAuley, J. Wang, and J. Li, "Optical wavelet transform classifier with positive real Fourier transform wavelets," *Optical Engineering* vol. 32(6), 1333-1339 (1993)

[16] X. J. Lu, A. Katz, E. G. Kanterakis, and N. P. Caviris, "Joint-transform correlator that uses wavelet transforms," *Opt. Lett.* vol. 17(23), 1700-1702 (1992)

[17] M. Sanghadasa, P. S. Erbach, C. C. Sung, W. A. Friday, "Wavelet transform applied to synthetic aperture radar - optical implementation and adaptive techniques," *Optical Engineering* vol. 33(7), 2282-2289 (1994)

[18] D. Roberge, and Y. Sheng, "Optical composite wavelet-matched filters," *Optical Engineering* vol. 33(7), 2290-2295 (1994)

[19] *Optical Engineering* vol. 31(9) (Sep. 1992)

[20] *Wavelet Applications*, Proc. SPIE 2242 (1994)

[21] *Optical Engineering* vol. 33(7) (July 1994)

[22] D. P. Casasent, and J-S. Smokelin, "Real, imaginary, and clutter Gabor filter fusion for detection with reduced false alarms," *Optical Engineering* vol. 33(7) 2255-2263 (1994)

[23] P. J. Burt, in *Multi-resolution Image Processing and Analysis*, A. Rosenfeld, ed., Chap. 2, Springer-Verlag: Berlin (1984)

[24] S. L. Tanimoto, "Template matching in pyramids," *Computer Graphics & Image Proc.* vol 16, 356-369 (1981)

[25] G. Eichmann, A. Kostrzewski, B. Ha, D. H. Kim, and Y. Li, "Parallel optical pyramidal image processing," *Optics Letters* vol 13, 431-433 (1988)

[26] S. P. Kozaitis, and W. E. Foor, "Binary optical correlation using pyramidal processing," *Optical Engineering* vol. 33(6) 1838-1844 (1994)

[27] Ph. Refregier, B. V. K. Vijaya Kumar, and C. Hendrix, "Multicriteria optimal binary amplitude phase-only filters," *J. O. S. A.* vol. 9(12) 2118-2125 (1992)

☆U.S. GOVERNMENT PRINTING OFFICE: 1995-610-126-50119

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.